

EAGLE
and the
High Level Architecture

Sponsor:

OSD PA&E

U.S. Army TRADOC Analysis Center

U.S. Army National Simulation Center

Jack Ogren

July 1996

MIT

Contexts

Eagle & the High Level Architecture

- Eagle Design Summary**
- Summary of Effort & Design**
- Technical Approach**
- Class Structures**
- Interactions**
- Event Synchronization**
- RTI Functionality Exercised**
- Testing - Results**
- Summary**

Contexts

Eagle & the High Level Architecture

Eagle Design Summary

- **Summary of Effort & Design**
- **Technical Approach**
- **Class Structures**
- **Interactions**
- **Event Synchronization**
- **RTI Functionality Exercised**
- **Testing - Results**
- **Summary**



Battle Design -- Summary

- **Used by TRAC as a combat development analysis tool to study corps and division level force effectiveness issues. (Not a Training Simulation)**
- **Characteristics**
 - **Corps & below level simulation**
 - **Resolution to Battalion or Company (Entity level w/BDSD)**
 - **Deterministic (Stochastic w/BDSD)**
 - **Hybrid event Structure**
 - = **Attrition/Ground Movement/Detection/C2 are Time-stepped**
 - = **Air/ADA interactions are Event-driven**
 - **Integrates Artificial Intelligence methods and conventional combat modeling algorithms**
 - Object-Oriented**
 - Embedded AI Systems (expert systems)**
 - Symbolic decision making**
 - Lisp/KEE Programming environment**
 - **Command and Control modeled explicitly.**
 - Units execute orders and pass information based on a Battlefield Management language.**

agle Design -- Summary

○ Functional Representation

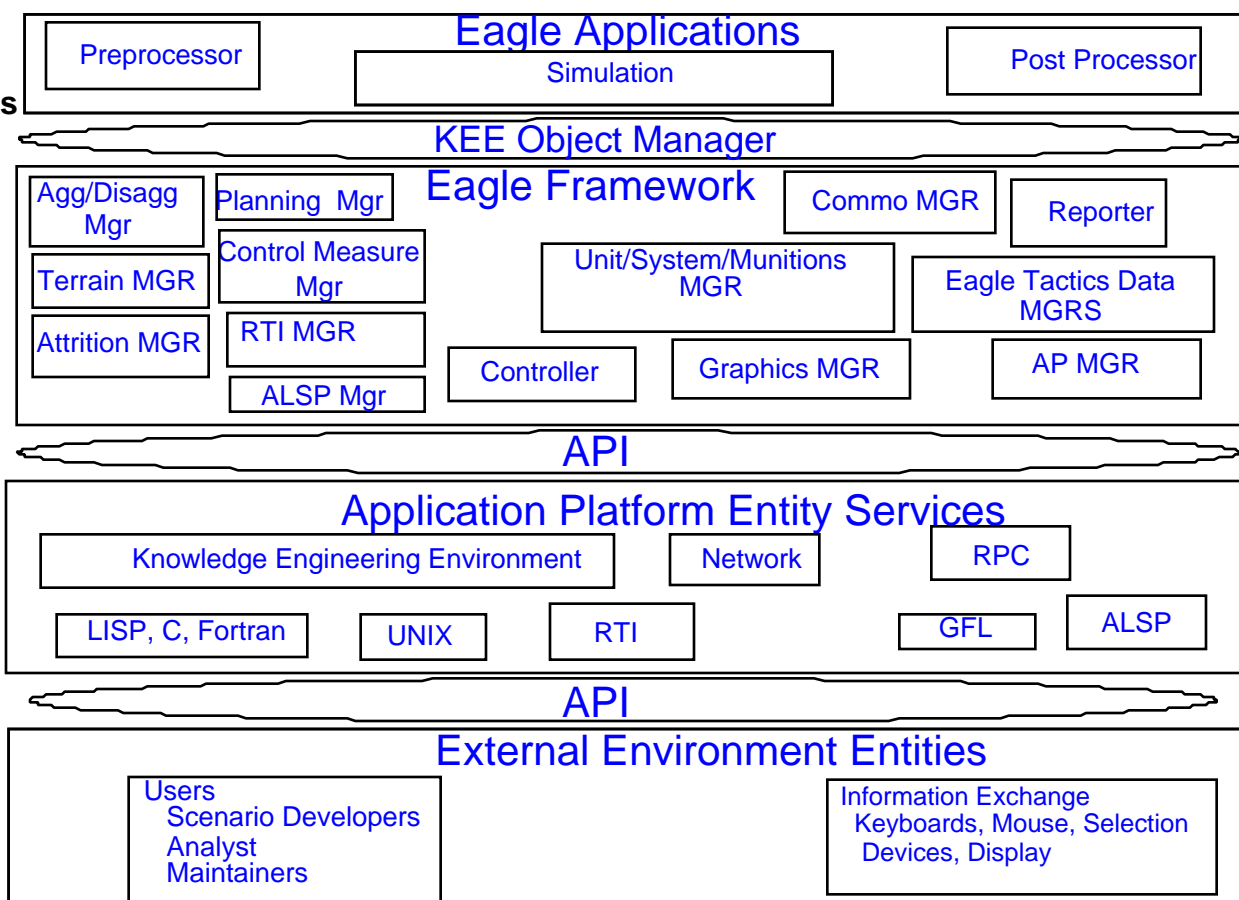
- **Command & Control Headquarters (Corps through Battalion)**
- **Ground Maneuver Units (Company, Battalion)**
- **Fire Support Units (Battalion, Battery, Tube, Radar section)**
- **Air Maneuver Units (Battalion, Company, Section)**
 - = **Attack and Scout Helicopter Units**
- **Air Defense Units (Battalions, Battery, Sections)**
- **Intelligence (Sections)**
 - = **Air & Ground Acquisition Elements (RPV, radar, JSTARS, RECCE)**
- **Engineers (Battalion, Team)**
 - = **Mobility, countermobility, Survivability**
- **Fixed Wing (Flights)**
 - = **CAS, BAI, SEAD**

Eagle Architecture

Applications are loaded individually or combined
Simulation = Combat Units

Eagle Framework are services provided to applications.
Responsible for maintaining ground truth.

Runs on HP and Sun workstations.



Contexts

Eagle & the High Level Architecture

- **Eagle Design Summary**
 - Summary of Effort & Design**
- **Technical Approach**
- **Class Structures**
- **Interactions**
- **Event Synchronization**
- **RTI Functionality Exercised**
- **Testing - Results**
- **Summary**

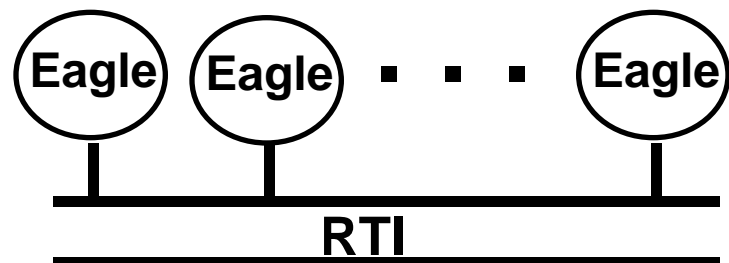


Eagle & the HLA

Summary of Effort to Date

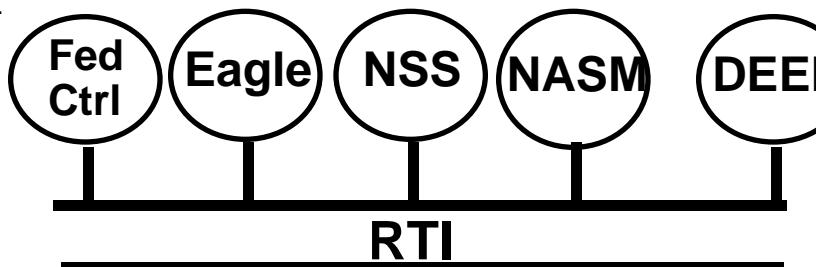
Eagle Early Analysis Experiment

Distributed Units -
Eagle Combat units interact
using the RTI.



Joint Training Federation (JTFp)

Distributed Functionality-
Army Combat units - Eagle
Navy Combat units - NSS
Air Force Combat units - NASM
Environment - DEEM
Federation Controller



Design facilitate: Running as Standalone
as Distributed Units
as Distributed Functionality
as Distributed Units & Functionality (not tested)

Distributed Eagle Goal & Basic Design

- **GOAL:** Decrease computation load on single processor while minimizing interactions and maintaining the same temporal, tactical, organizational and spacial consistency that currently exists on a single processor.
- **DESIGN:** Combat units are distributed among multiple Eagle simulations on the network.
 - = Each simulation maintains its own set of core services
terrain, terrain evaluation, attrition, tactics DB ... etc
 - = Each simulation reflects all units not owned by the simulation.
 - = The distributed design maintains consistency between:
Services on simulations (Terrain conditions).
Combat units and their reflected representations.

Functionally Distributed Eagle Goal & Basic Design

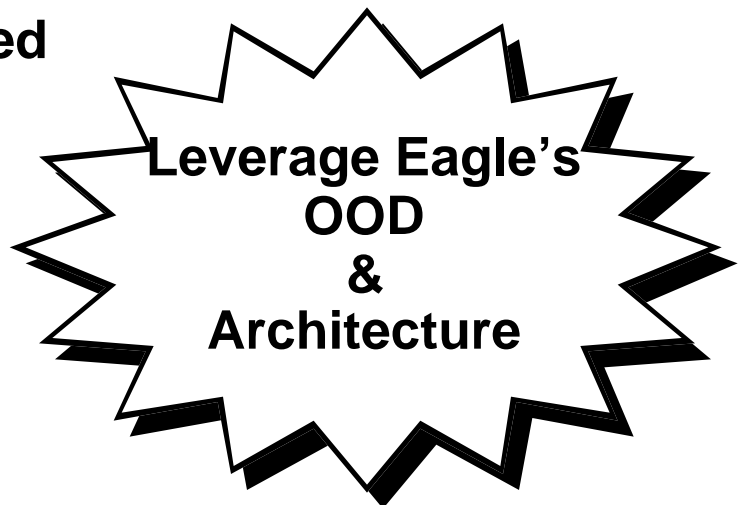
- **GOAL:** Allow externally generated Objects (combat or environmental) to interact with Eagle generated combat units while maintaining the same or better temporal, tactical, organizational and spacial consistency that currently exists on a single processor.
 - **DESIGN:**
 - Eagle provides all ground combat functionality for the federation. All ground combat units are reflected.
 - Functionality within Eagle is replaced by functionality provided by the confederation members.
 - = Eagle replaces normal fixed wing operations by
 - Subscribing to the federation air objects (classes)
 - Publishing & subscribing to interactions between the ground & air players (air to ground / ground to air)
-

Contexts

Eagle & the High Level Architecture

- Eagle Design Summary
- Summary of Effort & Design
 ↳ Technical Approach
- Class Structures
- Interactions
- Event Synchronization
- RTI Functionality Exercised
- Testing - Results
- Summary

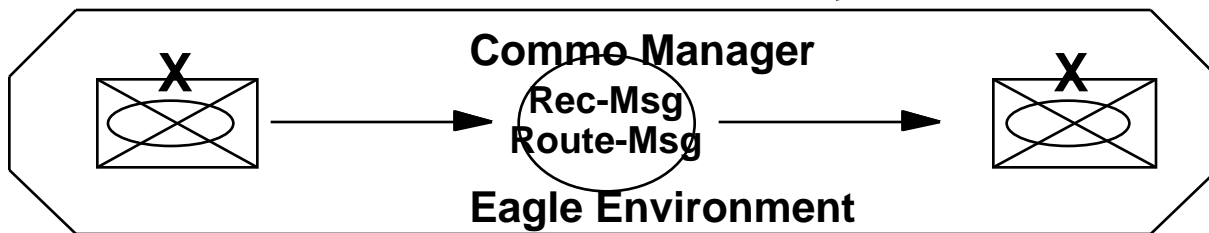
**{ Within Eagle
and
Between Eagle & the RTI**



Technical Approach within Eagle Example

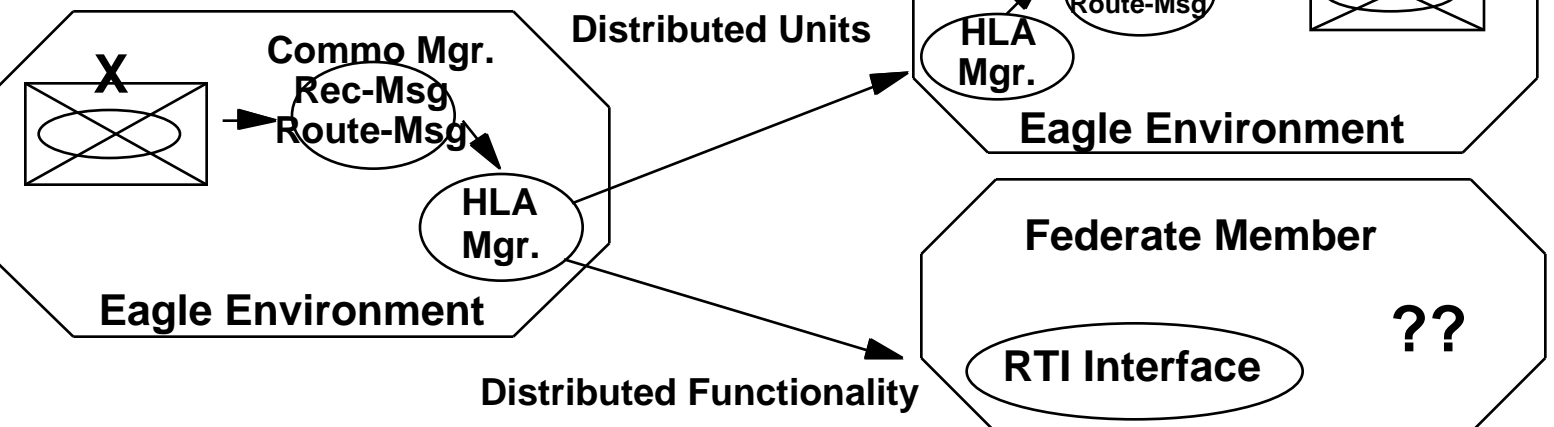
Communications between units.

Stand alone



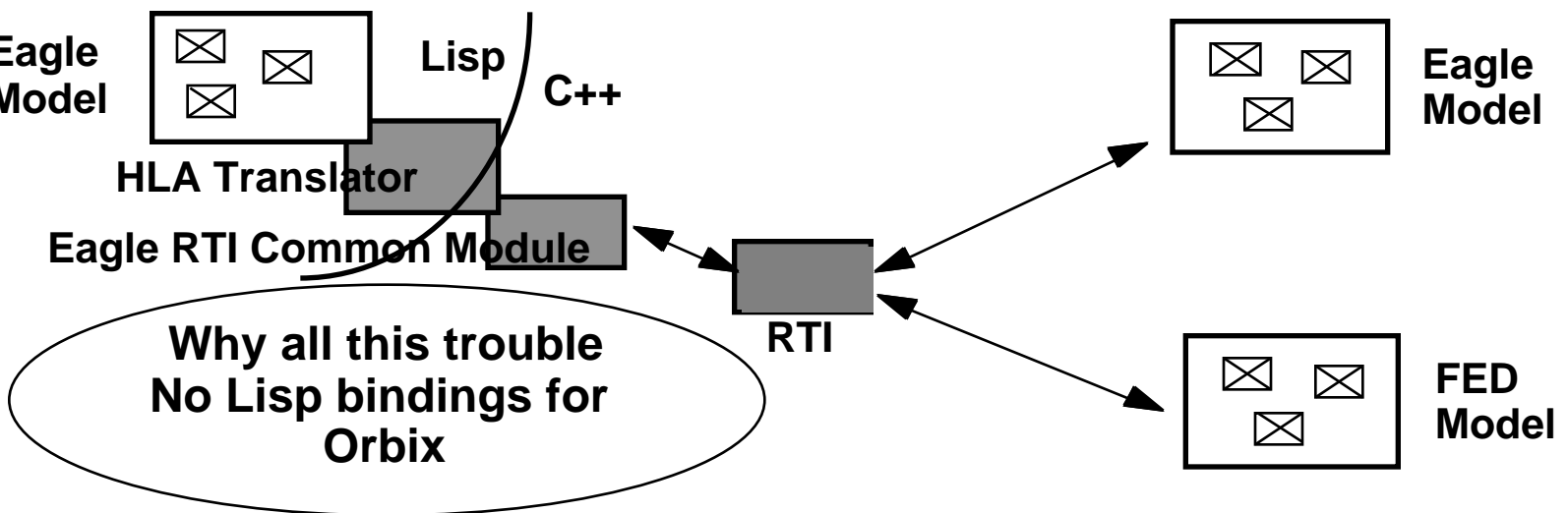
The Commo Mgr. delays msg
Jamming, Type equip, Combat state etc.

Distribute - where addressee is on different machine, else same as above.

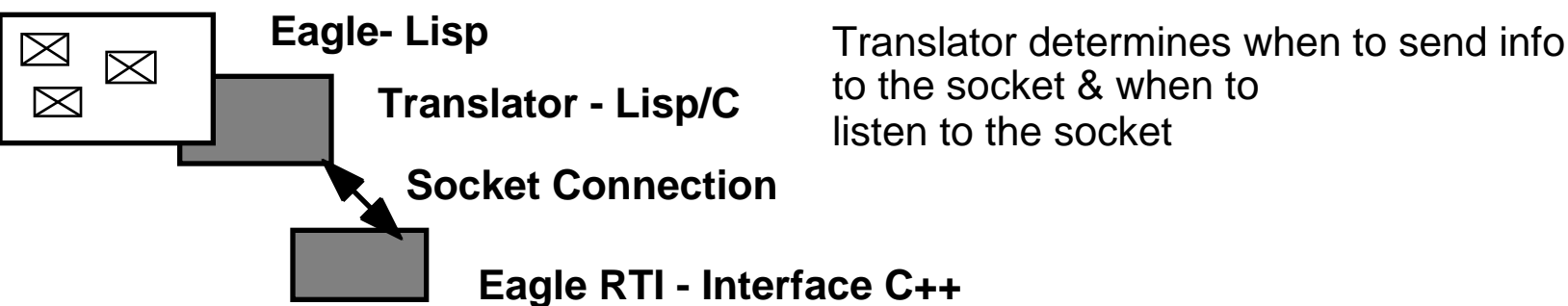


Technical Approach between Eagle & the RTI

- A new Eagle Service (HLA Translator) is provided within the Eagle framework to provide the interface between Eagle objects & the RTI.
- Each Eagle simulation HLA Translator Service (lisp) communicates with the RTI through an Eagle RTI Common module (C++).
- The Eagle RTI Common module provides the interface with the RTI - rti_ambassador, sim_ambassador, etc.



Technical Approach - External to Eagle



Translator - Lisp/C

Out the socket --> (hla_talk "TIME_ADVANCE_REQUEST^1.34")

Listen to socket (Blocked ?) --> (hla_listen)

Eagle RTI - Interface C++ Loop - Orbix Events or socket Calls -Do **SOCKET CALL**

Receive socket Stream: Tokenize, execute Case Statements which --

Calls rti_Ambassador: time_advance_request(the_time, IT_X)

Eagle RTI - Interface C++ Loop - Orbix Events or socket Calls -Do **ORBIX EVENT**

sim_amb_imp method is called: sim_ambassador_imp::time_advance_granted(Fed_time)

Outputs to socket "time_advance_granted 1.34"

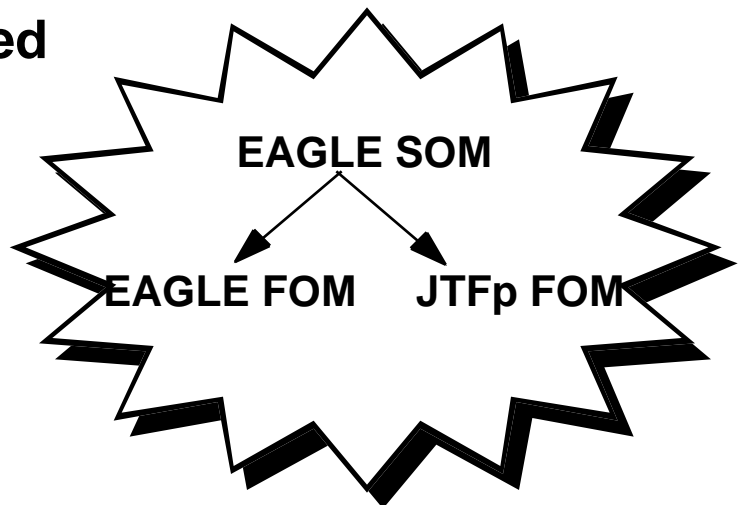
Calls to Sim_amb are buffered on Socket when Eagle is not Listening.

Contexts

Eagle & the High Level Architecture

- Eagle Design Summary
- Summary of Effort & Design
- Technical Approach
 - Class Structures
- Interactions
- Event Synchronization
- RTI Functionality Exercised
- Testing - Results
- Summary

**{ Software Object Model
and
Federation Object Model**



Eagle Class Structure

Distributed Eagle FOM

Eagle SOM

JTFp FOM



Distributed Eagle FOM (partial)

Base Class	1st SubClass	2nd SubClass
Military Units	Ground Movers	
	Air Movers	Air Maneuver
		Fixed Wing

JTFp FOM (partial)

Base Class	1st SubClass	2nd SubClass
Player	Air Player	Ballistic Missile
		Aircraft
		Flight
	Ground Player	Fixed Site
		Mobile Ground
		Aggregate Ground
	Float Player	

Eagle SOM Class Structure (partial)

Eagle SOM - Partial Class Structure				
MILITARY UNITS	AIR-MOVERS	FIXED-WING-UNITS	AIR-SENSOR-UNITS	BLUE-AIR-FLIGHT
				RED-AIR-FLIGHT
				BLUE-FLIGHT
				RED-FLIGHT
			FIXED-WING-ATTACKERS	
		HELICOPTER-UNITS	BLUE-HELO-BN	
			BLUE-HELO-CO	
			RED-HELO-BN	
			RED-HELO-CO	
			BLUE-AD-UNIT	
		AIR-DEFENSE-UNITS	RED-AD-UNIT	
			B-CBT-ENGR-TM	
		ENGINEER-UNITS	B-CBT-HVY-ENGR-TM	
			R-CBT-ENGR-TM	
			R-GP-ENGR-TM	
		ARTILLERY-UNITS	BLUE-ARTY-BN	
			BLUE-ARTY-BATTERY	
			RED-ARTY-BN	
			RED-ARTY-BATTERY	
		GROUND-SENSOR-UNITS	BLUE-GROUND-SENSOR-UNIT	
			RED-GROUND-SENSOR-UNIT	
		GROUND-MANEUVER-UNITS	BLUE-BN-TF	
			RED-MRR-BN	
			RED-TANK-BN	
			BLUE-CO	
			RED-CO	
		HQ-UNITS	BLUE-CORPS-CP	
			BLUE-DIV-CP	
			RED-ARMY-CP	
			RED-DIV-CP	
			BLUE-BDE-CP	
			BLUE-BN-CP	
			BLUE-ENGINEER-CP	
			RED-REGT-CP	
			RED-BN-CP	
			RED-ENGINEER-CP	

Class Structure

Attributes used to define Reflected Units

Distributed Eagle FOM

(Partial - Ground Mover)

CT	ATTRIBUTE
ARY-	Battlefield Operating System
	Higher Hdq's name
	Depth of unit (meters)
	Direction of movement of the unit
	Disaggregated boolean
	Echelon
	Percent Effective
	Effectiveness State
	Frontage of Unit (meters)
	Latitude
	Longitude
	Name
	Operational Activity
	Orientation of weapons
	Phantom boolean
	Purged boolean
	Quantity of Systems on-hand by type
	Route segment
	Side
	Size
	Systems requirements code
	System configuration
	Task
	Unit Type
ND-	Air Defense Controller
RS	Air Defense Network boolean
	Air Defense Status
	Assignment
	Command Assignment
	Command Unit boolean
	Indirect Fire Systems on-hand
	Number of Indirect Fire Units
	Prioritized Enemy List
	Signature - counter battery
	Signature - counter mortar
	Signature - combat
	Signature - communications
	Previously detected boolean
	Signature - IR
	Signature - Moving Target Radar
	Signature - photo
	Net speed from last time step
	System weight factor

JTFp FOM

(Partial - Aggregate
Ground Player)

OBJECT	ATTRIBUTE
Player	entity name
	federate id
	affiliation
	motion type
	voice nets
	itids nets
	trap tre
	comander type
	radar cross section
	radar detectable
Aggregate Entity	elint detectable
	comint detectable
	ir detectable
	photoint detectable
	air to air engageable
	surf to air engageable
	air to surf engageable
	surf to surf engageable
	composition
	time at last cse change
	lat at last cse change
	lng at last cse change
	alt at last cse change
	cse at last cse change
Dead Reckoned Players	hspd at last cse change
	vspd at last cse change
	depth
	front
Aggregate Ground Player	orientation

Number of Attributes to define
a typical ground combat unit in
Eagle: ~ 400

Number of Attributes to reflect
units in Distributed Eagle:

	Define	Update
Ground Movers	43	31
Air Maneuver	35	26
Fixed Wing	33	22

Number of Attributes to reflect
Aggregate Ground Units in JTFp:

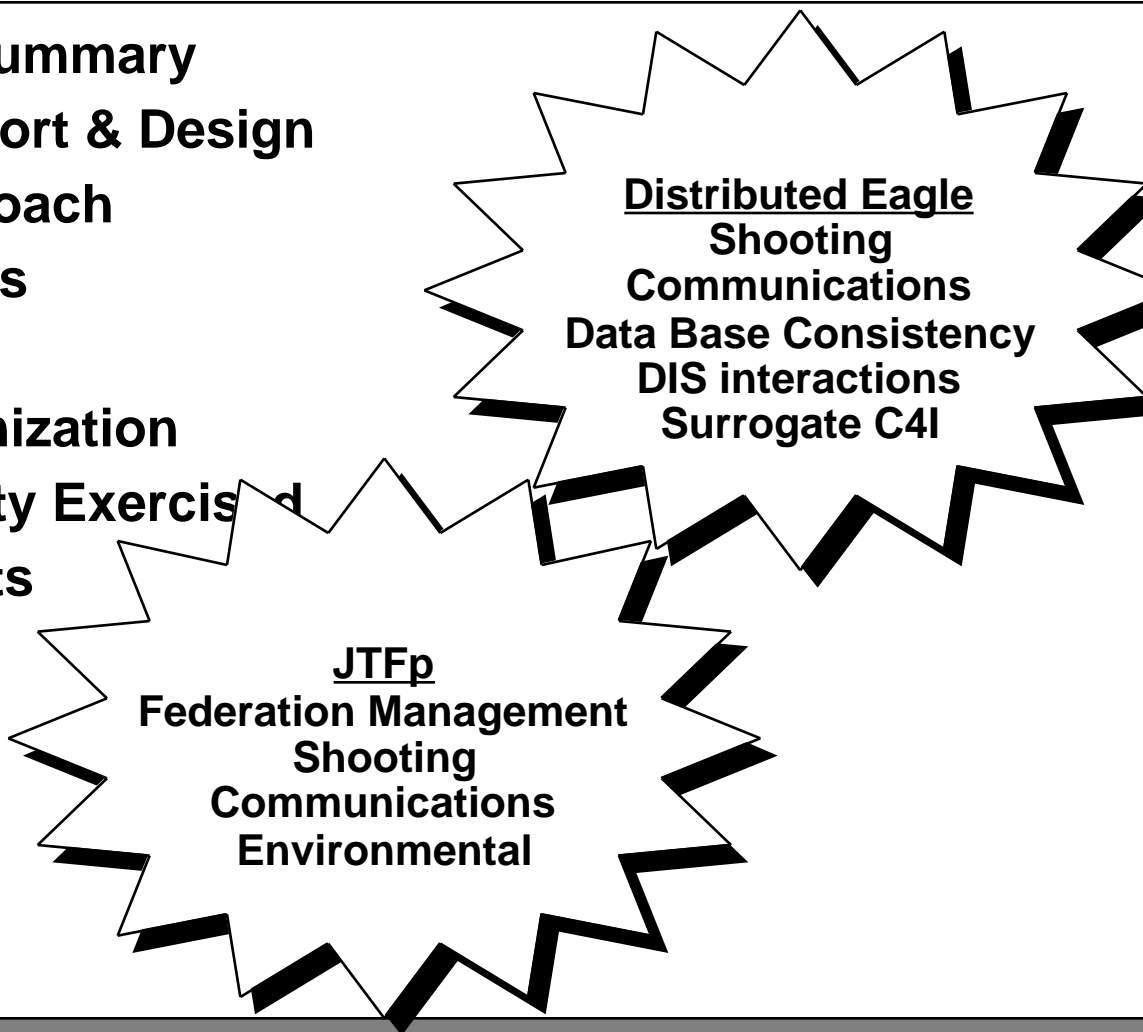
	Define	Update
Ground Movers	29	17

Attributes used for
detection's

Contexts

Eagle & the High Level Architecture

- Eagle Design Summary
- Summary of Effort & Design
- Technical Approach
- Class Structures
- Interactions
- Event Synchronization
- RTI Functionality Exercised
- Testing - Results
- Summary

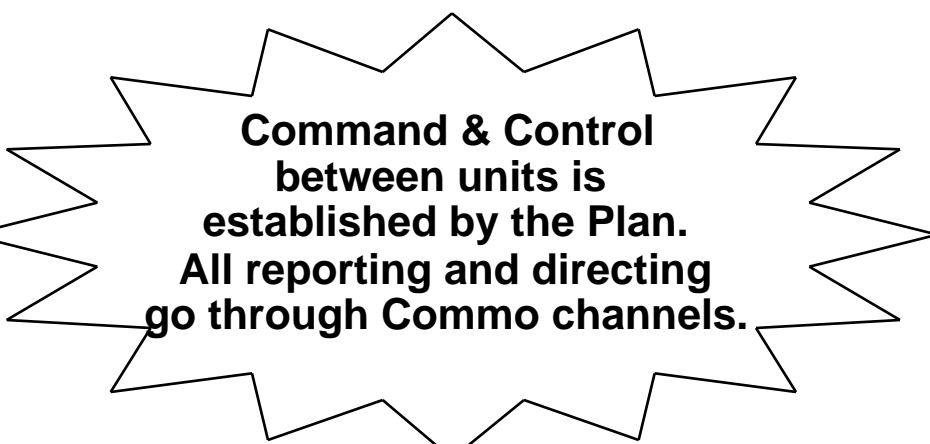


Distributed Eagle Interaction Structure

Typical Interactions between combat units and services

○ Unit Interactions

- Detection (implicit)
- Direct Fire
- Indirect Fire
- Air to ground
- Ground to air
- Communications



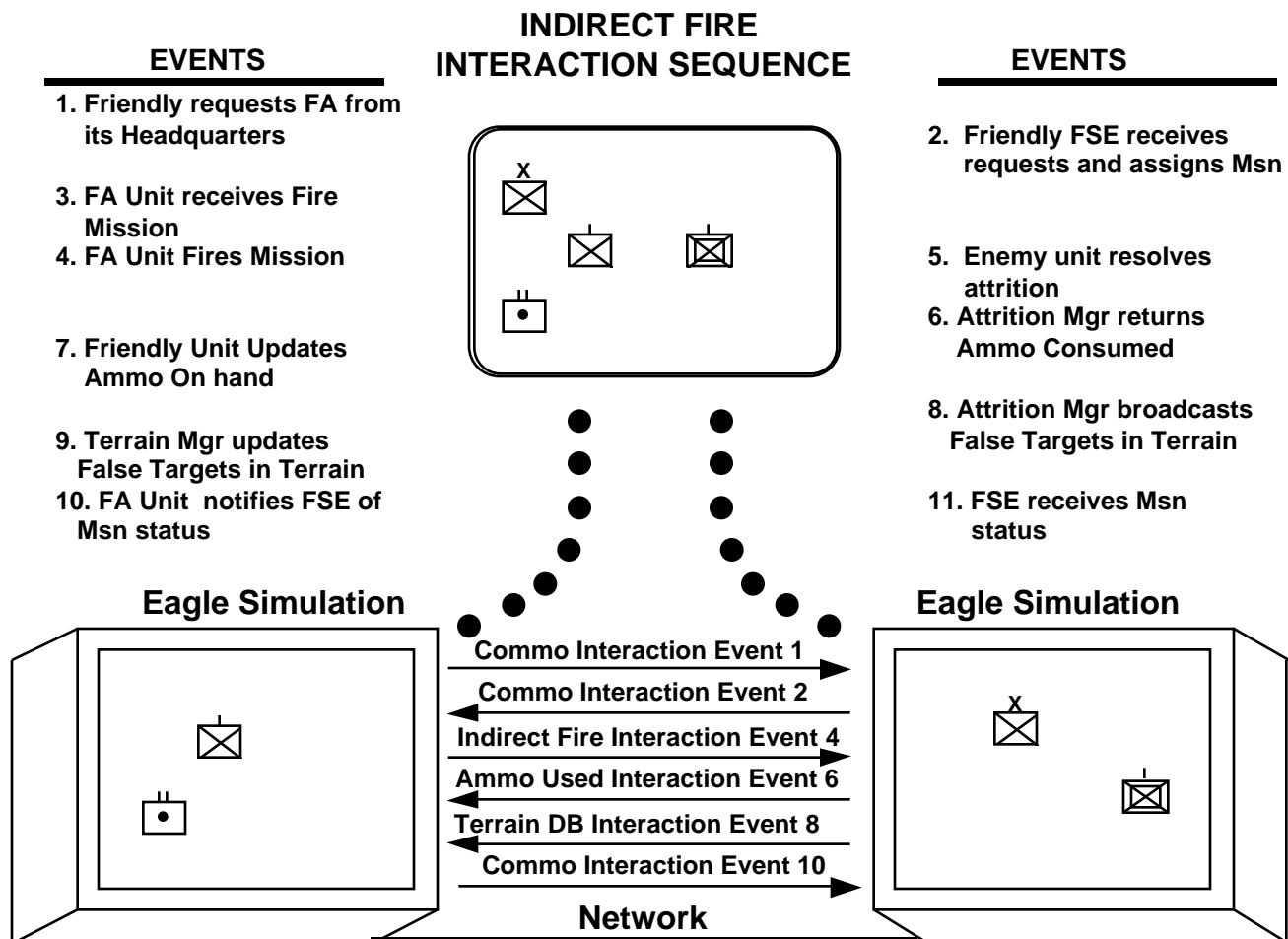
**Command & Control
between units is
established by the Plan.
All reporting and directing
go through Commo channels.**

○ Service Interactions

- Event Queue
Management
 - Add/remove events
- Control Measures
Database consistency
 - Create CM
- Terrain Database
consistency
 - false targets
 - register engr work
- Attrition
 - Ammo use
 - Suppression

Eagle Interactions - C2 ... Indirect Fire

Example Interactions in distributed Eagle



Current Interactions Allowed - Distributed Eagle

3 interaction types - total with subtypes: 65

Interaction	Initiating Class	Receiving Class
Engagement direct fire ground to ground	Attrition Manager	Attrition Manager
Bulk ammunition consumed	Attrition Manager	Ground-Movers, Air Movers
Engagement indirect fire ground to ground	Attrition Manager	Attrition Manager
Suppression update	Attrition Manager	Ground-Movers
Engagement Air to Ground	Attrition Manager	Ground-Movers
Aircraft abort flight notification.	Fixwing	Airspace manager
New air/ada events for Time Step.	Air Space Manager	Air Space Manager
Communications between units	Commo Manager	Commo Manager
Create control measure - db_consistency	Scenario Control Measures Manager	Scenario Control Measures Manager
Dead Target Updates - db_consistency	Attrition Manager	Terrain Manager
Register Eng. Work - db_consistency	Terrain Manager	Terrain Manager
Update Terrain Feature - db_consistency	Terrain Manager	Terrain Manager
Remove Terrain Feature - db_consistency	Terrain Manager	Terrain Manager
Ground receive losses air attack.	Attrition manager	Ground Movers
Activate IDF msn	Military Unit (Command)	Attrition Manager
Update tf occupany - db_consistency	Terrain Manager	Terrain Manager
Update-active-aois	Ground-Mover	Air-Maneuver
Create-breach - db_consistency	Terrain Manager	Terrain Manager
Create-bypass - db_consistency	Terrain Manager	Terrain Manager
Set Unit visibility terrain- db_consistency	Military-units	Terrain Manager
Set Unit visibility map- db_consistency	Military-units	Terrain Manager
Set unit visible - db_consistency	Military-units	Military-units
Db-consistency, used to coordinate events	Aggregate actors	Aggregate actors
Eagle Management - stop, start ...	Eagle Controller, Confederation Manager	Eagle Controller, Confederation Manager
SIU - Dis interactions between actors	Model Network Manager	Model Network Manager
Resolution Unit Interface - C4I interactions	C4I Interface, Military Unit	C4I Interface, Military Unit
CommandUnit Interface - C4I interactions	C4I Interface, Military Unit	C4I Interface, Military Unit

JTFp Interactions that involve Eagle

○ Unit Interactions

- Detection (implicit)
- Indirect Fire
- Air to surface
- Surface to air
- Communications

○ Federation Management Interactions

- Initialization
- Environmental
 - LOS

Interaction	Initiating	Receiving
TBM Warming	Player	Eagle Commo Manager - Player
TBMLaunchAlert	Player	Eagle Commo Manager- Player
Situation Report	Player - Eagle Commo Manager	Player (JTF HQ)
RequestAirSupport	Player - Eagle Commo Manager	Player (JTF HQ)
AirToAggregate GroundEngage	Player	Eagle Attrition Manager
DiscreteGroundToAirEngage	Player	Eagle Attrition Manager
AggregateGround ToAirEngage	Eagle Attrition Manager	Player
Get & Return LOS	Eagle Terrain Manager	Surface Cover
Initialize Federation	Federation Status	Eagle Federation Status
Execute Federation	Federation Status	Eagle Federation Status

**JTFp has defined 24 Interactions
Eagle participates in 10**

Contexts

Eagle & the High Level Architecture

- Eagle Design Summary
- Summary of Effort & Design
- Technical Approach
- Class Structures
- Interactions
 - ◀ Event Synchronization
- RTI Functionality Exercised
- Testing - Results
- Summary



The Challenge

To maintain a consistent view of time within Eagle regardless of the types of simulations that are participating in a federation.

Simulations that use RTI Time Mgt

- 1. Use time steps to advance time**
timesteps > Eagles (DEEM)
timesteps < Eagles
timesteps = Eagles (Eagle)
- 2. Use events to advance time (NSS,NASM)**

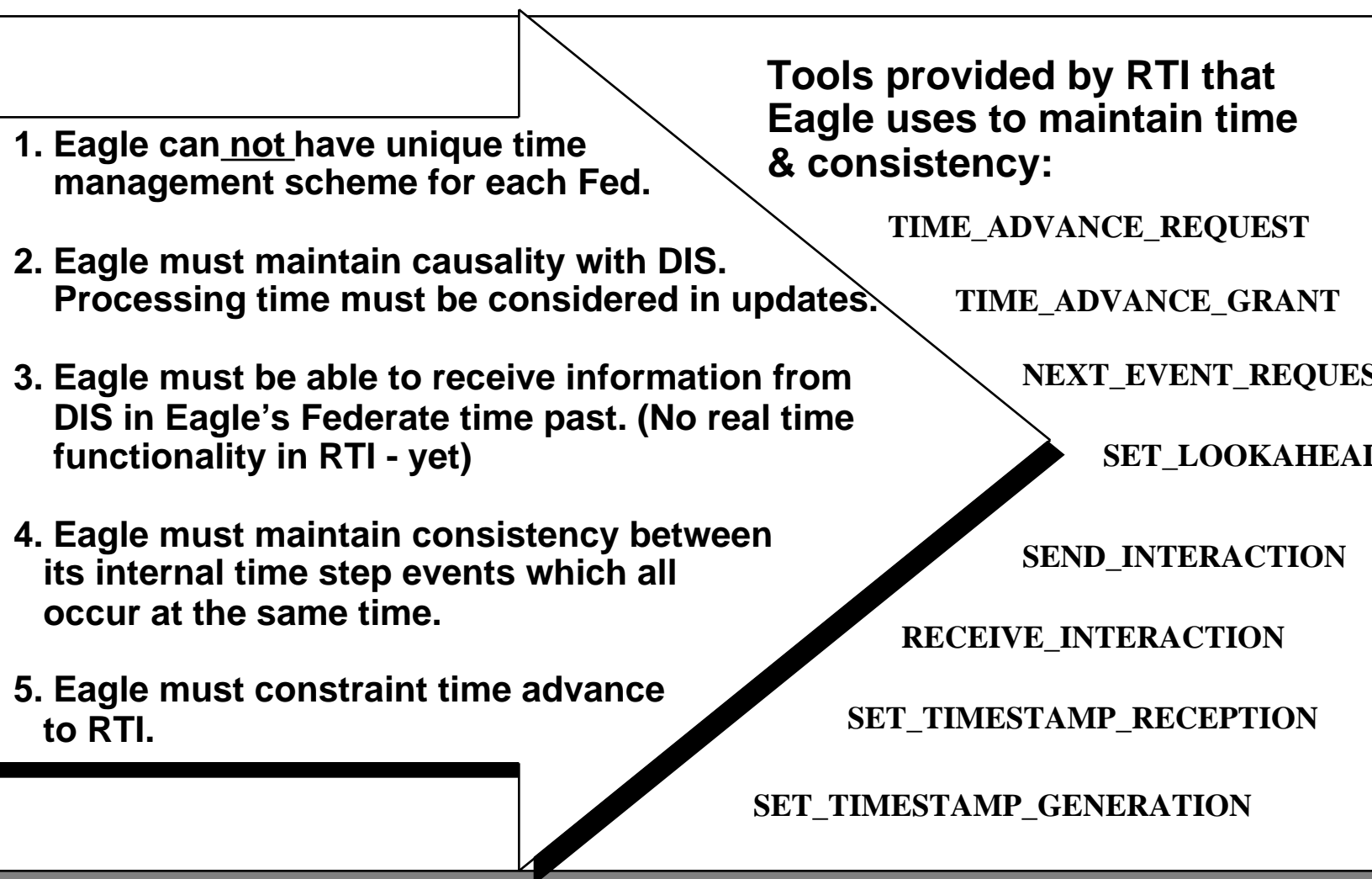
**Running as fast as possible
or scaled real time**

**Simulations that do not
use RTI Time Mgt (DIS)**

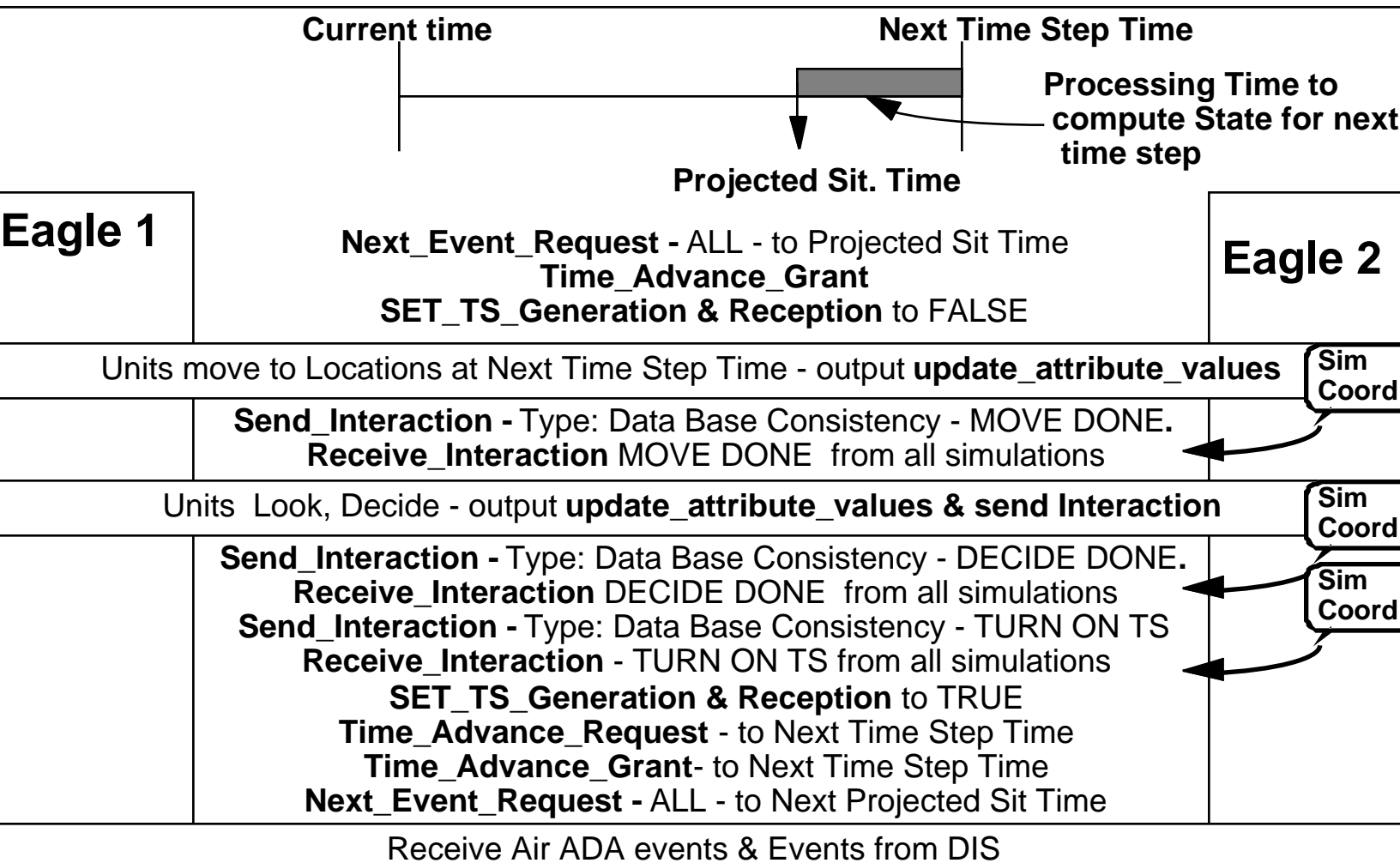
Real time

Eagle has a hybrid event structure that relies on both the notion of continuous time using time steps (1 to 5 mins) and the projecting of discrete events limited to the duration of the time step.

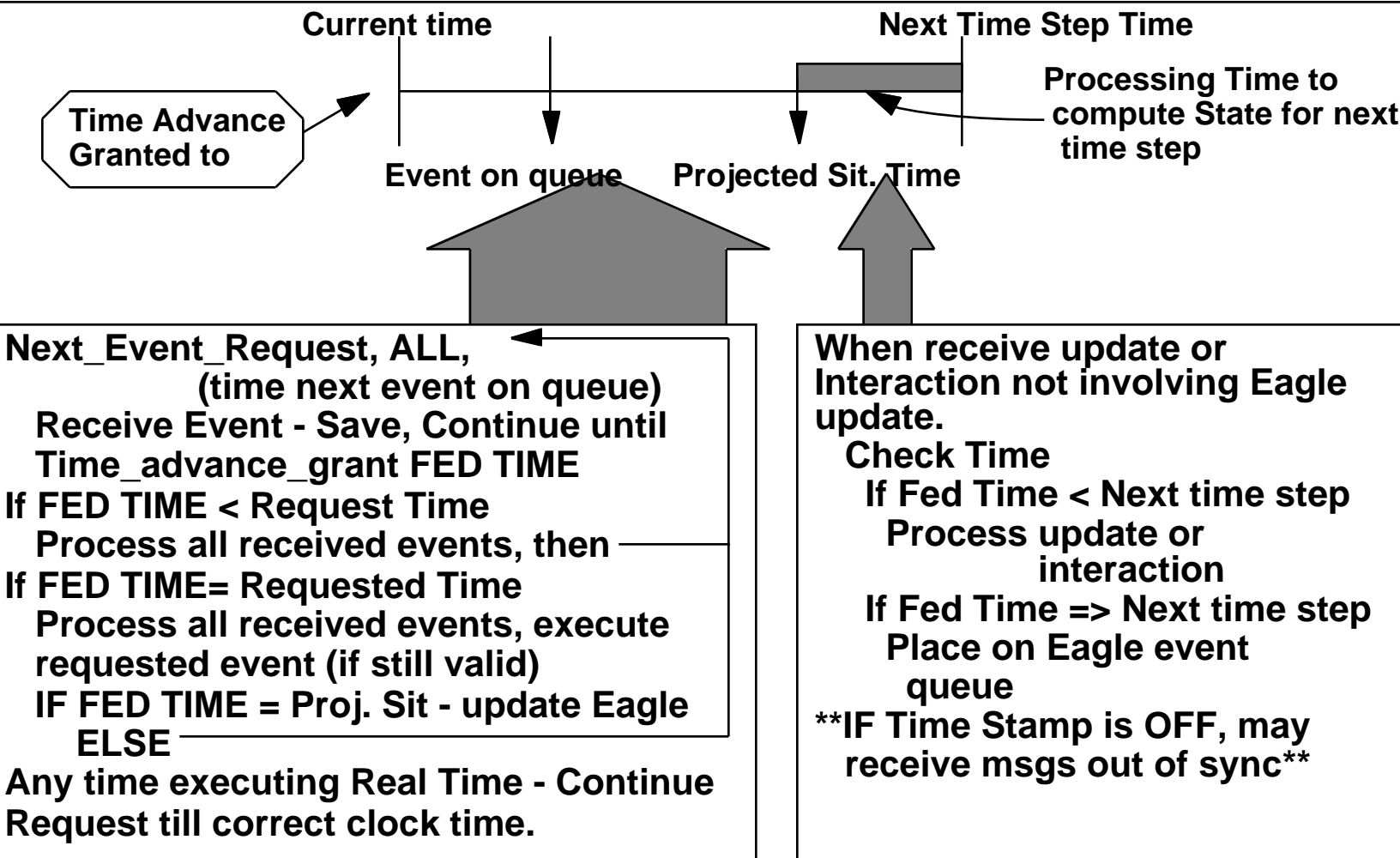
Additional Challenges



DISTRIBUTED EAGLE



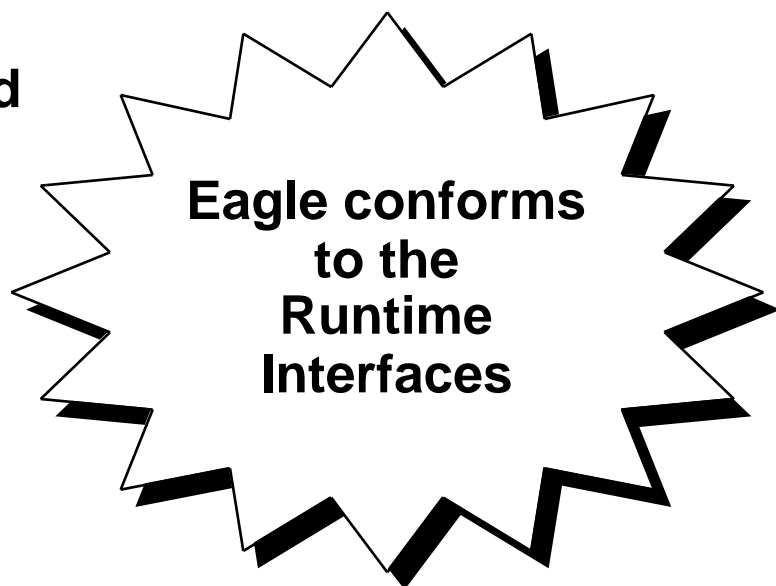
DISTRIBUTED EAGLE W/EVENTS



Contexts

Eagle & the High Level Architecture

- **Eagle Design Summary**
- **Summary of Effort & Design**
- **Technical Approach**
- **Class Structures**
- **Interactions**
- **Event Synchronization**
- **RTI Functionality Exercised**
- **Testing - Results**
- **Summary**



RTI - Functional Interfaces Used Distributed & Functional Area

Type	Avail	Used	Type	Avail	Used
Federation Management	22	3	Ownership Management	19	1
Declararation Management	12	4	Object Management	20	15
Time Management	17	6			

Federation Management

CREATE_FEDERATION_EXECUTION
 JOIN_FEDERATION_EXECUTION
 RESIGN_FEDERATION_EXECUTION

Declararation Management

PUBLISH_OBJECT_CLASS
 UNSUBSCRIBE_OBJECT_CLASS
 PUBLISH_INTERACTION_CLASS
 UNSUBSCRIBE_INTERACTION_CLASS

Time Management

TIME_ADVANCE_REQUEST
 NEXT_EVENT_REQUEST
 GET_TIMESTAMP_GENERATION
 GET_TIMESTAMP_RECEPTION
 GET_LOOKAHEAD
 TIME_ADVANCE_GRANT

Ownership Management

UNCONDITIONAL_ATTRIBUTE_OWNERSHIP_DIVESTITURE

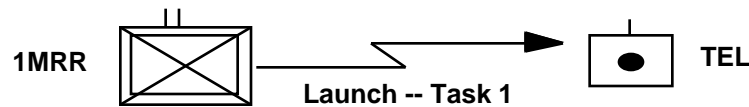
Object Management

ID_REQUEST
 INSTITUTE_OBJECT
 TRANSLATE_ATTRIBUTE_NAME
 TRANSLATE_INTERACTION_CLASS_NAME
 TRANSLATE_OBJECT_CLASS_NAME
 DELETE_OBJECT
 UPDATE_ATTRIBUTE_VALUES
 SEND_INTERACTION
 SEND_DIRECTED_INTERACTION
 INSTITUTE_DISCOVERED_OBJECT
 REMOVE_OBJECT
 REFLECT_ATTRIBUTE_VALUES
 RECEIVE_INTERACTION
 RECEIVE_DIRECTED_INTERACTION
 PROVIDE_ATTRIBUTE_VALUES

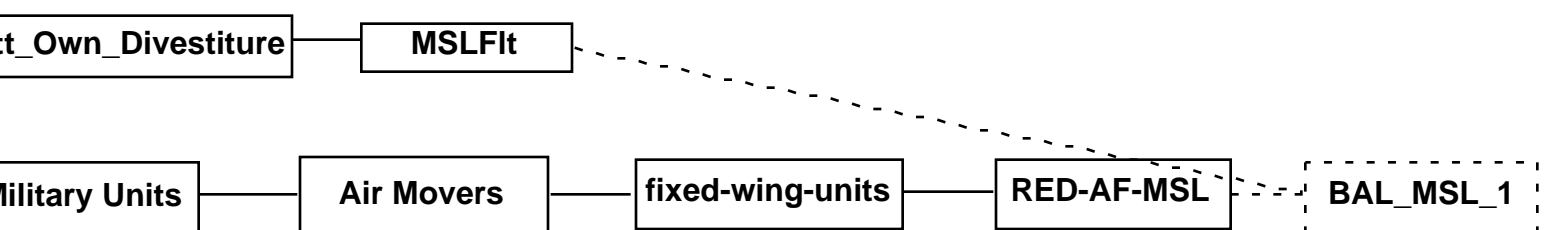
The Above are used &
understood
 by Eagle

Functional Interfaces Ownership Management

Eagle creates and NASM Flies



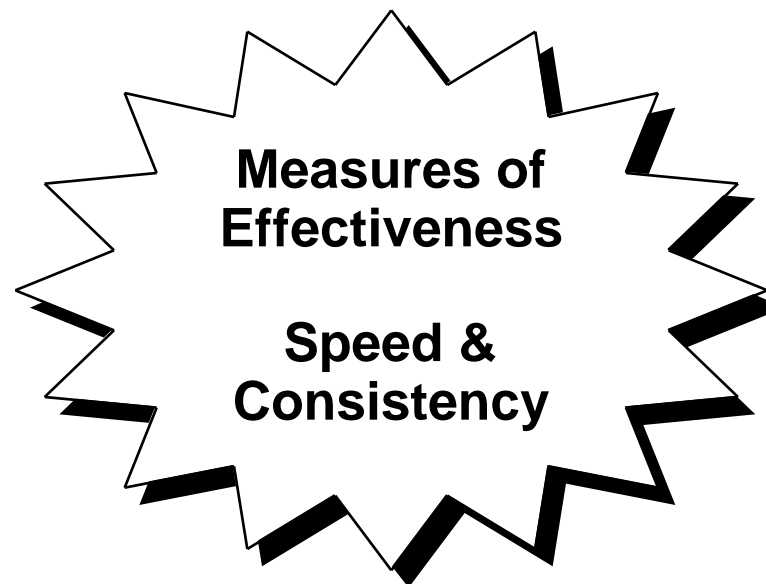
1. TEL receives msg to Launch Ballistic Missile - Changes to Operational Activity Pre Launch
 - Create Ballistic Missile - Eagle Class RED-AF-MSL with name "bal_msl_1"
 - Initialize with Operational Activity SpinUptoLaunch, determine Launch Time
 - If Func-area-distributed
 - Id_request, Instantiate_Object, Update_attribute_values**
 - If Not Func-area-distributed place launch event on ASM event queue
 - If Func-area-distributed
 - A. Place Divest-Msl-flight event on ASM event queue
 - B. **Unconditional_attribute_ownership_divestiture**
 - To NASM - location & Time of arrival attributes
2. Air Space Manger processes Divest-Msl-flight event
 - Attach Bal_msl_1 to Class MSLFlt. This will cause methods to be inherited by BAL_MSL_1 that will over write normal move & update methods, allowing all movement of the MSL to be caused by updates from the Federation.



Contexts

Eagle & the High Level Architecture

- **Eagle Design Summary**
- **Summary of Effort & Design**
- **Technical Approach**
- **Class Structures**
- **Interactions**
- **Event Synchronization**
- **RTI Functionality Exercised**
- **Testing - Results**
- **Summary**



Results - Distributed Eagle

Consistency & Time

- Not much news so far - TIME
- Consistency - Simple Test 2 machines

		Machine 1	Machine 2
OUT	Type	Num	Num
	FED	1	2
	DEC	33	33
	OBJ	355	343
	OBJ OUT	8007	7816
	OWN	0	0
	TIME	723	723
	TOTAL	9119	8917
IN	FED	0	0
	DEC	5	5
	OBJ	7816	8007
	OWN	0	0
	TIME	144	144
	TOTAL	7965	8156

1. No Messages lost.
2. Duplicates
Ver .30 avg: 100
Ver .33 none
3. Consistency
with standalone
??

Time - 2 machine run 100 units/machine

ALSP: 21 to 1
HLA : 24 to 1



Data point
of 1 ..so

**BUT IT
WORKS**

Contexts

Eagle & the High Level Architecture

- **Eagle Design Summary**
- **Summary of Effort & Design**
- **Technical Approach**
- **Class Structures**
- **Interactions**
- **Event Synchronization**
- **RTI Functionality Exercised**
- **Testing - Results**
 - **Summary**

Summary

○ Continue Testing

- Increase number of machines
- Increase number of units
- Increase run time
- Increase interactions



Measures of
effectiveness
TIME
CONSISTENCY

○ Version .33U seems capable to support continued testing.



HOWEVER

I CAN SAY NOW THAT

**The underlying concepts of the HLA
support the distribution requirements
of a constructive analytical model.**